



전투 시스템의 대들보

(The Fundamental Pillars of a Combat System)

작성자: 세바스찬 람보틴(Sebastien Lambottin)

작성일: 2012년 8월 15일

전투 시스템(combat system) 설계는 많은 게임 디자이너들이 자주 반복해서 하게 되는 작업이다. 아주 오래 전 일이지만 내가 처음 전투시스템 설계를 해야 했을 때 꽤 힘들었다. 설계에서 좋은 결과를 얻기도 힘들었을 뿐만 아니라, 노련한 전투 디자이너들이 흥미진진한 전투 메카닉을 만들 때 사용하는 공식을 설명하는 정보를 찾기도 힘들었다.

전략적 계획의 예시



나는 다른 기획자들의 도움을 받아 전투 시스템 설계의 기본적인 룰에 대한 지식을 수집하고 공식화(formalize)하기로 했다. 이 글은 그 노력의 결과다.

우리가 전투 시스템의 게임플레이 메카닉을 설계할 때 생각하는 주요 목표는 '플레이어가 현명한 선택을 하여 적절한 타이밍에 적절한 능력(ability)을 사용하도록 유도하는 것'이다. 우리는 플레이어가 다음에 취할 액션을 예상하고 전투에 있어서 전략적 계획을 개발할 수 있기를 바란다.

이런 결과를 얻을 수 있는 방법은 여러 가지가 있겠지만, 전투 시스템에서 플레이어의 능력을 설계하는데 도움이 되는 아주 중요한 특성 두 가지를 소개하자면 다음과 같다.

- 각각의 능력은 독특한 기능을 한다. 특정 영역을 타격하거나, 적의 스톤(stun)을 유발시키거나...
- 각각의 능력은 사용시의 리워드(reward)와 리스크(risk)가 균형을 이루고 있다.

전술적 능력

다음은 이런 특성을 염두에 두고 <콜 오브 듀티(Call of Duty)> 같은 고전 밀리터리 슈팅 게임의 능력들을 분석한 사례이다.

1. 각각의 능력은 독특한 기능을 한다

이런 능력들을 설계하는 또다른 방법은 각각을 플레이어의 도구로 생각하는 것이다.

다음 도식은 <콜 오브 듀티>의 능력들이 영향을 끼치는 범위를 보여준다.

콜 오브 듀티의 타격영역 예시



- **근접공격(Melee attack):** 플레이어 정면의 가까운 거리를 커버한다. 근거리에서만 이 기술을 쓸 수 있지만 한 방으로 적을 죽일 수 있다.
- **일반 사격(Normal shot):** 플레이어가 언제든지 사용할 수 있는 기본 공격 방법. 중거리일 때 최적의 공격방법이다.
- **조준 사격(Iron sight shot):** 멀리 있는 적에 대해 매우 정밀한 슈팅(예. 머리를 쏜다거나)을 할 때 적합하다. 주변을 볼 수 없기 때문에 매우 위험하다.
- **수류탄(Grenade):** 엄호물이나 코너에 있는 적을 공격할 때 적당하며, 한방에 적을 죽일 수 있다. 소비성이기 때문에 꼭 필요할 때만 사용하도록 주의해야 한다.

이제 플레이어가 이런 능력들을 즉시 실행할 수 있다고 생각해보자. 재미있는 도전과제가 나타나는데, 바로 '적절한 시점에 적절한 버튼을 누른다'는 것이다.

하지만 이걸로 충분치 않다. 앞서 언급한 것처럼 우리가 전투 시스템을 설계할 때 진짜 목적은 플레이어의 영리함과 전투 전략에 도전하는 것이다. 따라서 우리는

기본적으로 다양한 선택이 가능한 시스템을 만들고, 플레이어가 각각의 상황에서 가장 적절한 옵션을 선택하게 할 것이다.

2. 전술측면- 각각의 능력에 리스크/리워드의 균형 유지

각각의 능력은 특정한 방식의 공격을 가능하게 할 뿐 아니라, 나름의 장점과 단점도 있다. 여기서는 <스트리트 파이터 2(Street Fighter II)>의 예로 자세히 살펴보자.



기획자는 다양한 방식으로 한 가지 능력의 장점과 단점을 조합할 수 있다. 액션 게임에서 가장 일반적으로 사용되는 사례를 몇 가지 소개해 본다.

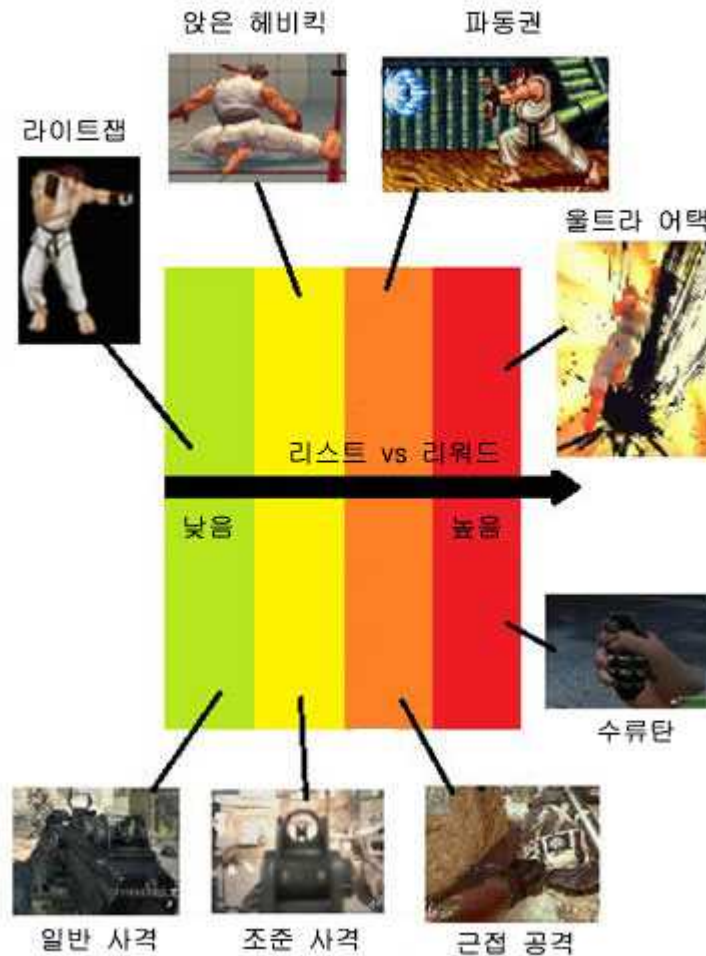
장점: 데미지 정도, 스톤, 격퇴(Repel), 시간에 따라 커지는 데미지, 눈멀게 하기, HP 리젠

단점: 포인트 소비, 회복, 발동시간, 회복시간

각각의 능력들이 리스크와 리워드 간에 완벽하게 균형을 이루었다고 하더라도, 리스크 대비 리워드 계수가 다른 능력들을 플레이어에게 보여주는 것이 좋다.

예컨대 플레이어의 어떤 능력은 적에게 데미지를 조금밖에 입힐 수 없지만 빠르기 때문에 리스크가 적다. 정 반대로 큰 데미지를 줄 수 있지만 준비나 회복시간 탓에 매우 위험할 수 있는 특별한 공격방법도 있다.

리스크 vs. 리워드 밸런스



각각의 전투 시스템은 리스크 대 리워드 계수가 다른 여러 가지 능력들을 제공한다

또, 같은 능력이 적을 방어하는 도구로 쓰이기도 한다. 플레이어의 능력들이 적을 공격할 때뿐 아니라 적의 공격을 맞받아칠 때(counter attack)에도 쓰일 수 있다는 것을 기억해두자. 따라서 플레이어가 어떤 능력을 적절치 않은 시점에 쓴다면 적을 공격할 기회를 놓치는 것은 물론이고 적의 공격에 대응하지 못해서 HP 를 잃을 수도 있다. 이 때문에 플레이어가 능력을 실행할 때 갖게 되는 리스크 가운데 카운터 어택(counter attack)도 들어가게 된다.

세 가지 도전

액션 게임이 실시간이 되면서 플레이어가 상황의 변수를 파악하고 사용할 능력을 결정해야 하는 빈도도 늘어났다.

거리 판단

- 적과의 거리를 측정하여 어떤 능력을 사용할 것인지 정한다.
- 그 능력을 수행한 후에 캐릭터 위치가 어디가 될 것인지를 판단한다.

모든 것이 실시간으로 움직이기 때문이다.

시간 측정

- 능력을 수행하는데 필요한 시간을 측정한다.
- 그 능력이 지속되는 시간을 예상한다.

모든 것이 시간에 따라 변화하기 때문이다.

영리함과 예비

- 여러 가지 상황에서 수행할 액션의 조합을 예비해 둔다.
- 적의 공격에 대응하는데 사용할 능력이 무엇인지 안다.

각각의 능력이 플레이어의 전술적 도구이기 때문이다.

이와 동시에 적들을 플레이어의 능력에 맞추어 기획할 때엔 많은 이점이 있다. 사실, 적을 추가하는 주요 목적 중에 하나가 플레이어에게 게임 메카닉을 조금씩 가르치기 위함이다.

적은 플레이어를 위한 도전과제

우리가 인공지능(AI) 행동을 설계할 때는 보통 AI가 가능한 영리하고 사람처럼 느껴지게 하려는 경향이 있다. 예를 들자면 다수의 적이 분대로 움직이게 하거나, 플레이어가 똑똑하다고 생각할 행동을 하게 만든다.

하지만 AI 가 아무리 영리하게 행동하더라도, 플레이어가 이 적을 마주했을 때 경험하게 되는 도전(challenge)이 게임에 가장 큰 영향을 준다는 사실을 우리는 쉽게 잊어버린다. 다시 말해 게임 디자이너로서 우리는 도전을 설계하는데 근본이 되는 것이 무엇인지 되돌아볼 필요가 있다. 그래서 우리가 목표로 하는 도전을 제공하기 위해 가장 중요한 AI 속성이 무엇인지 생각해보아야 한다.

각 적이 제공하는 도전을 세밀하게 정의한다. 적의 주요 기능이 플레이어를 공격하는 것이고, 또 대부분의 경우 플레이어가 적을 무찌를 수 있기 때문에, 나는 적의 행동을 설계할 때 다음의 두 가지 중요한 질문을 해본다.

- 플레이어는 어떻게 적을 공격하고 무찌를 수 있는가?
- 플레이어는 어떻게 적으로부터 자신을 방어할 수 있는가?

사실, 한 무리의 적을 기획할 때 우리는 각각의 적을 무찌를 다양한 방법을 만들려고 노력한다. 이 점을 설명하기 위해 <스파이더맨 샷터드 디멘션즈(Spider-Man: Shattered Dimensions)>의 두 가지 예를 살펴보자.



플레이어는 어떻게 닌자를 방어하는가?

플레이어가 닌자의 공격을 피하기 위해서는
2초 범위 내에 닷지(dodge)를 해야 한다

닌자의 공격



플레이어의 방어



플레이어가 닷지를 너무
일찍 하면 닷지 후 돌아오는
동안에 타격을 받는다.

플레이어가 너무 늦게
피하면 타격을 받기 때문에
타격애니메이션 중간에
닷지를 시작할 수 없다





플레이어는 어떻게 카미카제
бом버를 방어하는가?

카미카제 bomber 공격



플레이어를 향해 달림

폭발 후
죽음



플레이어의 방어



카미카제를 쏘거나 때릴 시점

대응하기 늦음

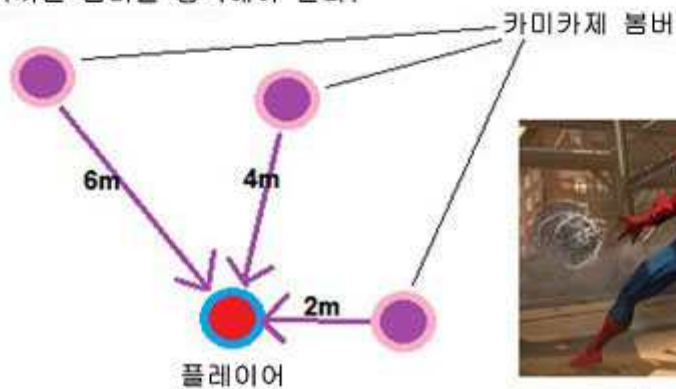


бом버가 플레이어를 향해 달려오는 동안은
공격 받기 쉽고 약하다. 한방에 죽일 수 있다.



бом버가 폭발하면
폭발로 플레이어가 다친다.

다수의 bomber를 만났을 때: 여러 bomber가 플레이어를
공격할 때 재미있어진다. 플레이어는 돌아서서 가장
가까운 bomber를 공격해야 한다.



위의 도표에서 보듯 플레이어가 닌자와 카미카제 bomber의 공격을 방어할 때는 각기
다른 도전 과제가 생긴다.

플레이어가 닌자로부터 자신을 방어하는 가장 좋은 기술(능력)은 **닷지(dodge)**이다.

플레이어가 닌자로부터 자신을 방어할 때 도전과제는 **타이밍(timing)**이다.

플레이어가 카미카제 폭탄으로부터 자신을 방어하는 가장 좋은 기술(능력)은 **웹샷(webshot)**이다.

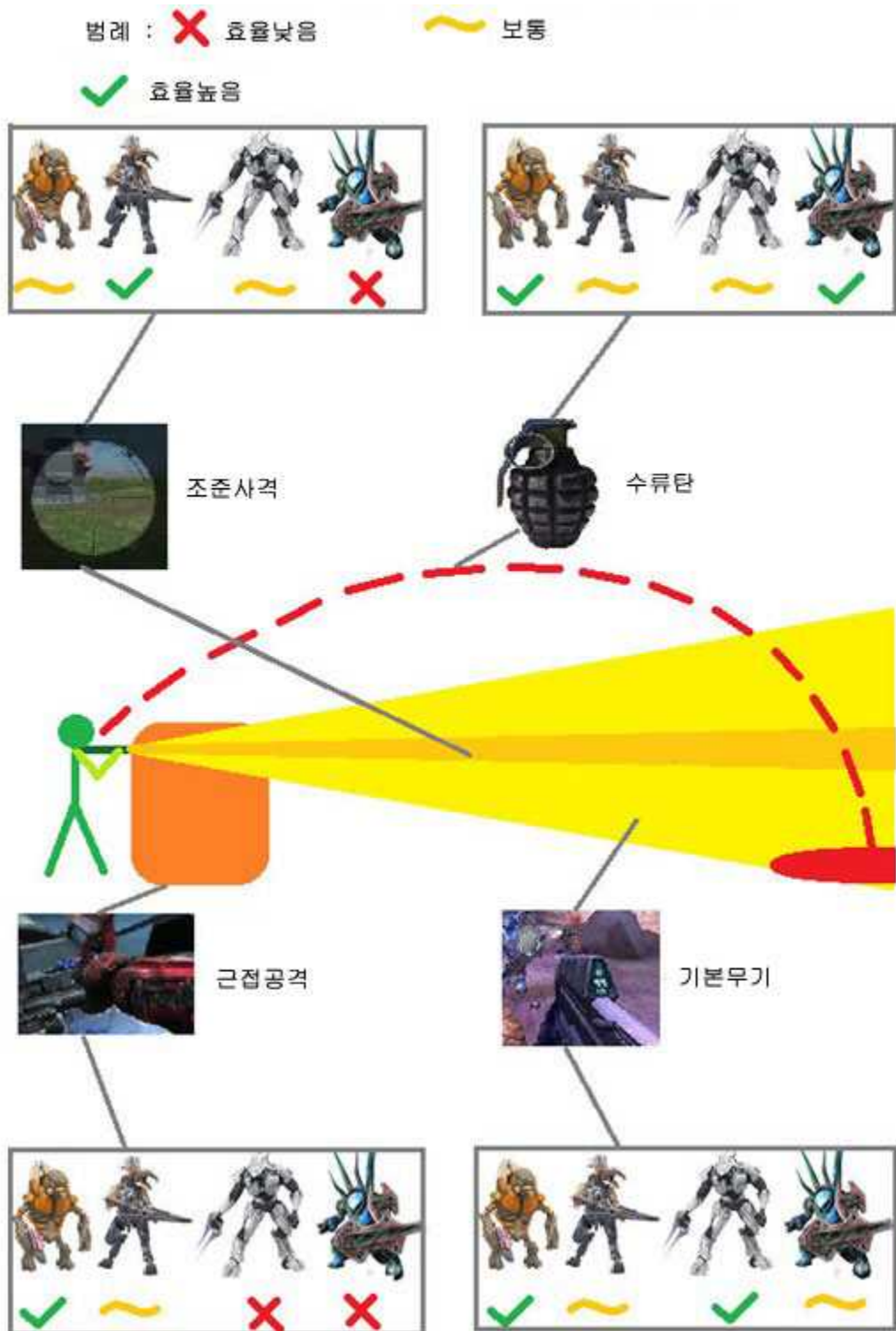
플레이어가 카미카제 폭탄으로부터 자신을 방어할 때 도전과제는 **거리 판단(distance evaluation)**이다.

플레이어 능력에 도전

적들마다 각각 서로 다른 도전과제를 갖게 하는데 더하여, 능력/무기에 대한 약점을 각기 다르게 설계하는 것도 재미있다. 다시 말해, 적에 따라 각각의 무기의 효율이 달라지게 기획하는 것이다.

- 이렇게 하면 플레이어가 보유한 능력을 모두 사용하게 유도하는가? - 플레이어가 각 무기의 특성을 이해하는데 도움이 되는가? - 플레이어가 전략적으로 행동하고 적절한 능력을 적시에 사용하도록 유도하는가?

이 점을 설명하기 위해 <헤일로(Halo)>에서 적에 따른 무기 효율성 분석의 예를 살펴보자.



이런 속성 탓에, 플레이어는 여러 종류의 적과 전투를 벌일 때 이 능력 저 능력을 번갈아 사용해 더 효과적으로 전투에 임하게 된다.

액션 게임에서 공통적인 적의 원형

모두가 적을 플레이어의 능력과 매치하려고 하기 때문에, 여러 다양한 액션 게임에서 같은 종류의 적들이 나타나는 경우가 많다. 사실, 이런 공통적인 원형을 사용하면 많은 이점을 누릴 수가 있다.

- 도전과제를 쉽게 이해할 수 있다.
- 알아보기 쉽게 구분이 되는 형태를 띠고 있다.
- 튜토리얼 없이도 플레이어가 이들을 대적하는 방법을 알 수 있다.

액션게임에서 가장 일반적으로 나타나는 전형적인 적들과 이들이 게임 속에서 가지는 기능은 다음과 같다.

- 방패를 든 적: 기본 공격을 정확하게 하기



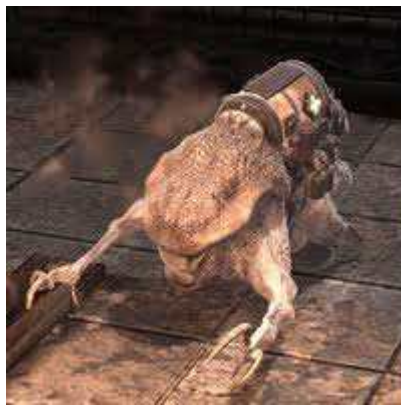
- 중무장한 적: 강한 공격(heavy attack)으로 파괴하기



- 스나이퍼: 장거리 공격



- 봄버(bomber): 근거리 공격으로 대응하기



원형의 서브클래스와 베리에이션

우리는 적들을 몇 가지 원형으로 분류하여 조직하고 싶어하면서도 한편으론 다양한 적을 보여주어 게임 전체의 도전 수위를 높이고 싶어한다.

서브클래스(sub-class)는 플레이어들이 이미 배운 룰을 이용해 플레이하면서도 적의 새로운 능력을 맞아 - 예컨대 기회의 틈이 좁아진다면, 더 정확하게 해야 한다든지 - 전투 시스템의 한계까지 계속 플레이하게 하는 역할을 한다.

우리가 적을 설계할 때는 적들 각각의 유익한 속성과 까다로운 속성을 분석하여 도전과제를 생각할 수 있다.

유익한 속성: 유리한 속성을 가진 적은 플레이어가 다른 적에게 데미지를 주는데 이용할 수 있다.

까다로운 속성: 까다로운 속성을 가진 적은 무찌르기가 좀 더 복잡하다.

<마리오(Mario)>게임에서 서브 클래스 컨셉의 예를 살펴보자.

본래의 적 원형



- 정해진 경로를 따라 플레이어를 향해 단순히 앞으로 이동한다.
- 플레이어가 적의 위로 점프하면 없앨 수 있다.
- 적을 없앤 뒤 나오는 거북껍질로 다른 적들을 죽일 수 있다.

서브 클래스의 원형



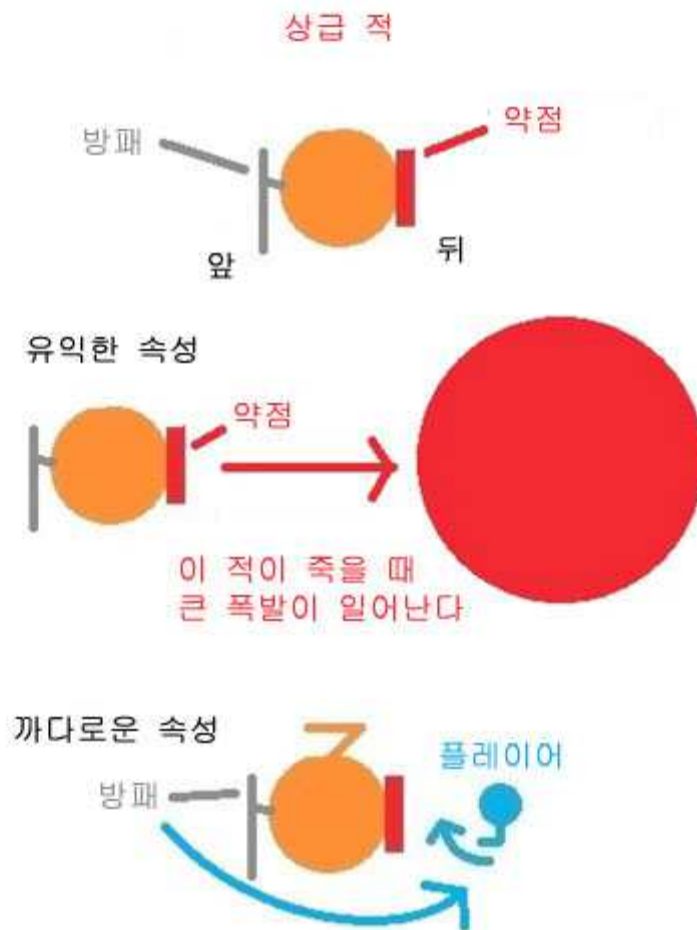
- 플레이어를 향해 앞으로 이동한다.
- 이 서브클래스의 까다로운 속성은 날아서 이동한다는 점이다.
- 거북껍질로는(역시) 다른 적들을 죽일 수 있다.

서브클래스를 설계하기에 좋은 방법은 한 클래스의 정해진 룰을 바꾸지 않은 상태로 난이도를 높이거나 낮추는 것이다.

리스크와 리워드 관리 사례:

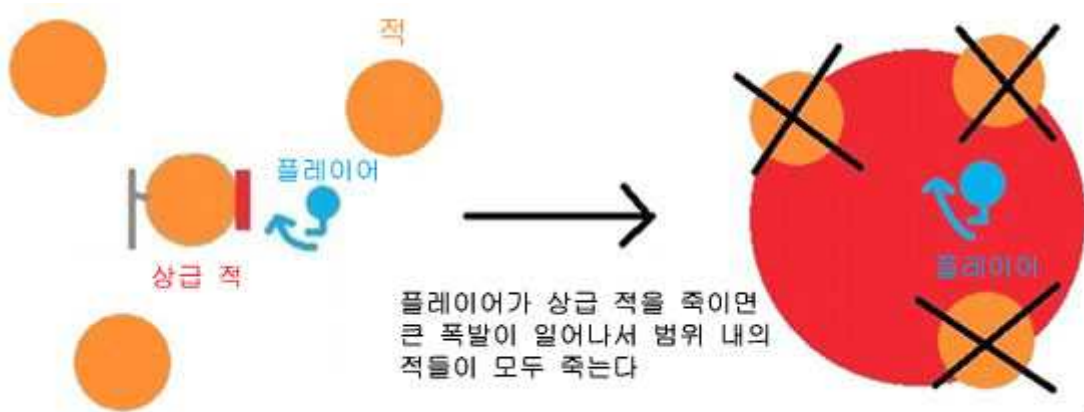
전투에 또다른 전략을 도입하기 위해서 서브클래스를 이용해 같은 적에 유익하고 까다로운 속성을 추가할 수 있다.

방패를 들고 등에 약점을 가진 상급 적이 있다. 플레이어가 등의 약점을 여러 번 가격하면 폭발한다.



적에게 방패를 주어서 죽이기 어렵게 만들었다. 플레이어가 뒤로 돌아가서 싸야지만 데미지를 입힐 수 있기 때문이다. 하지만 등 쪽에 약점을 추가하면서

플레이어에게 전략적 기회도 주었다. 보통의 경우에는 가장 쉬운 적부터 죽여서 살아있는 적의 수를 가능한 빨리 줄이는 것이 좋다. 하지만 이 경우에는..



다른 적들이 아직 살아 있는 상태에서 상급 적을 먼저 죽이려고 하는 것은 위험도가 높다. 하지만 동시에 약점을 이용해 적 전체를 더 빨리 죽일 수가 있다. 리스크를 어떻게 처리할지는 플레이어에 달렸고, 플레이어가 원하는 보상 정도에 따라 달라진다.

결론

이 글에서 기억해두어야 할 요점은 다음과 같다.

잘 설계된 전투 시스템은 플레이어가 몇 시간 동안 즐길 수 있는 시스템이다. 전투 시스템이 잘 구성된 게임은 매우 다양한 전투를 제공할 것이다.

이런 훌륭한 전투 시스템을 만들어내는 좋은 방법은 우선 플레이어 캐릭터에 다양한 능력을 설계하는 것이다.

플레이어의 능력은 다양한 방식으로 차별화될 수 있다. 예컨대 기획자가 특별한 속성-스턴, 리젠, 시간에 따른 데미지, 쫓기 등-을 추가할 수 있다.

두 번째로 기억해 둘 것은 플레이어의 능력과 매치하여 적을 설계하는 것이다. 각각의 적들은 특별한 도전과제를 제시해서 플레이어가 적을 물리치는데 특정한 능력을 사용하도록 유도해야 한다.

플레이어가 비디오 게임에 가장 몰입하는 경우는 스스로가 똑똑하다고 느끼고 자신의 영리함에 감탄할 때다. 그리고 전투 시스템은 플레이어가 그런 기분을 느끼도록 하는 훌륭한 도구이다.